Journal of Intelligent and Robotic Systems **28:** 125–149, 2000.

Invariant Representation and Matching of Space Curves

CHONG-HUAH LO
Telcordia Technology, RRC 1S327, 444 Hoes Lane, Piscataway, NJ 08855, U.S.A.

HON-SON DON*
Department of Electrical Engineering, National Chung–Hsing University, Taichung, Taiwan, ROC
(Author for correspondence)*

(Received: 6 November 1998; in final form: 24 August 1999)

Abstract. Space curves are highly descriptive features for 3-D objects. Two invariant representations for space curves are discussed in this paper. One represents space curves by complex waveforms. The other represents space curves using the 3-D moment invariants of the data points on the curves. Space curve matching using invariant global features is discussed. An algorithm for matching partially occluded 3-D curves is also presented, in which rigidity constraints on pairwise curve segments are used to determine the globally consistent matching. An association graph can be constructed from the local matches. The maximal cliques of the graph will determine the visible part of the model curves in the scene. Experimental results using 3-D curves obtained from stereo matching and edges detected from the range data are also presented.

Key words: space curve, invariant representation, matching, 3-D moment invariant, Fourier descriptor, Legendre function.

1. Introduction

Because of the recent development of range sensing techniques, many researchers have proposed robust object recognition and positioning algorithms using 3-D information [6]. In order that the computers can efficiently recognize objects in the scene, the vast amount of noisy image data is generally compressed into highly descriptive and discriminative features. The space curves (3-D curves) are important 3-D features, since they are rich in information and can be easily stored and manipulated. The 3-D coordinates of curves can be obtained from edge detection on range data or from matching of stereo images, etc. Edges which correspond to discontinuities of depth or surface orientation are not sensitive to changes of viewpoint. Space curves also appear in some computer representations of 3-D objects, such as the generalized cone and cylinder representation of elongated objects [1, 20]. Computer recognition of such objects is to match the spine axes and the junctions of the axes with those of the models. The curves are distinct from other geometric entities in that they have an invariant and intrinsic parametrization by

their own arc length. Volumes and surfaces do not have such a natural parametrization. This parametrization of space curves provides a natural ordering of points on the curves, and a natural correspondence mapping between two curves. Therefore, in curve matching problems, no more computation is required to find the point correspondence than that to determine the relative offset between two curves.

Space curves can be either represented by the intersection of two surfaces, or by the parametric equation $(x(t), y(t), z(t))$. Neither of these two representations is invariant under translation and rotation of the coordinate system. Invariant representation of the 3-D objects is important in pattern recognition systems. Closed contours have been represented by a set of invariant features constructed from the elliptic Fourier coefficients of the contours [13, 14]. On the other hand, Mokhtarian [18], and Kehtarnavaz and de Figueiredo [11] proposed to use the curvature and torsion functions to represent space curves. Their approach is a straightforward extension of the curvature representation of 2-D curves to 3-D curves. The scalar curvature functions of 2-D curves have been used to guide the curve segmentation [23, 27] and to compute global [34] and local features [3, 22] for curve matchings. The scale-space images [30] of the curvature functions have been used in a syntactic approach to 2-D shape analysis [19]. However, the curvature and torsion representation of 3-D curves can not be used for many computer vision applications, because the torsion function can not be defined at a point where the curvature vanishes. In other words, the representation of a curve becomes meaningless when the curve contains straight line segments.

In this paper, we propose an invariant waveform representation of the 3-D space curves. The waveform is represented by a complex function. The magnitude of the waveform is the nonnegative curvature function of the space curve and the phase of the waveform depends on the torsion of the curve. Thus, the waveform vanishes wherever the curvature of the curve vanishes. The phase of a complex function is undefined at a point where the function vanishes. This invariant waveform is unambiguously defined for any (smooth) curve appearing in the computer vision problems. It contains the complete 3-D information of the curve. Invariant features can be extracted from the waveform. Criterion functions which depend on the waveform and its derivatives can be defined for the decomposition of the curve into smooth segments separated by breakpoints. Using this representation, the segmentation and recognition of 3-D curves can be reduced to the problem of segmentation and recognition of 1-D complex waveforms. The scale-space description of the complex waveform can also be derived when it is expanded by convolution with a Gaussian filter. We found that the fingerprint theorem [31, 32] can still be used to reconstruct the complex waveform from its scale-space image. The waveform representation of space curves provides a framework that allows many statistical and syntactic pattern analysis techniques for planar curves and 1-D signals to be extended to 3-D curves. Another invariant representation of a space curve is to describe it by a feature vector of invariant 3-D moment functions computed from the curve or its breakpoints [16, 24]. This representation is straightforward and the

invariant moment features are robust in pattern recognition. Moment computation does not require estimation of derivatives from noisy data. So it is less sensitive to noise than the first representation.

When curves are represented by vectors of invariant global features, the statistical pattern classification techniques can be applied for the recognition. The recognition scheme will identify a given curve with a model curve. The relative orientation and translation between these two curves can then be computed by the method of least mean square estimation. We will show that the point correspondence between the curves can be quickly determined by matching the waveforms of the two curves.

If the curves on the object are partially occluded, their global features can not be used for pattern classification. We propose a template/structural curve matching algorithm for recognizing the partially occluded space curves. The curves in the scene are divided into segments which are used as templates. Each template is locally matched to various portions of the model curves. Rigidity constraint can be applied to pairwise curve segments to obtain a globally consistent matching. An association graph is constructed from the matched pairs. The maximal set of structurally compatible matches is determined by the maximal cliques in the association graph. Each clique corresponds to one model curve, and determines the visible part of the model curves in the scene. The proposed waveform representation can be used for fast template matching. A short and preliminary version of this paper was presented in [17].

The paper is organized as follows. In Section 2, we give a brief discussion of the differential geometry of the space curves, and the derivation of the invariant representation of the space curves. In Section 3, we propose an algorithm for recognizing space curves using the global features. We then discuss the determination of the point correspondence and the coordinate transformation. In Section 4, we discuss the template/structural approach to recognize the partially occluded curves. The experimental and simulation results are given in Section 5. Section 6 concludes the paper with a summary of results and a few remarks.

2. Invariant Representation of Space Curves

A regular space curve segment is a vector-valued function $\boldsymbol{x}\colon [a, b] \to \mathbb{R}^3$. The derivatives of $\boldsymbol{x}$ exist and are continuous up through order k ($k \geqslant 1$) for all t in $[a, b]$. A regular space curve segment can always be reparametrized by the arc length parameter. A curve parametrized by its arc length is called a unit speed curve. In this paper, we shall assume that all space curves are unit speed curves. The curvature function $\kappa(s)$ of a unit speed curve is the magnitude of the vector field $\mathrm{d}\boldsymbol{T}/\mathrm{d}s$ along the curve, where $\boldsymbol{T}$ is the tangent vector field, and $\kappa(s) = \sqrt{(x'')^2 + (y'')^2 + (z'')^2}$. The principal normal vector field $\boldsymbol{N}(s)$ is defined by the

direction of $\mathrm{d}\boldsymbol{T}/\mathrm{d}s$ at each point on the curve. The binormal vector field is defined as [10, 21]

$$\boldsymbol{B}(s) = \boldsymbol{T}(s) \times \boldsymbol{N}(s). \tag{1}$$

The torsion function is defined by

$$\tau(s) = -\big\langle \boldsymbol{B}'(s), \boldsymbol{N}(s) \big\rangle. \tag{2}$$

The triple $[\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{B}]$ forms a local orthonormal system at each point of the space curve. The normal vector, binormal vector and torsion can be unambiguously defined only at those points where the curvatures do not vanish. A straight line in 3-D space has a vanishing curvature function, so its torsion function can not be defined. The evolution of $[\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{B}]$ along the curve is governed by the Frenet–Serret equation

$$\big[\boldsymbol{T}'(s), \boldsymbol{N}'(s), \boldsymbol{B}'(s)\big]^{\mathrm{T}} = A\big[\boldsymbol{T}(s), \boldsymbol{N}(s), \boldsymbol{B}(s)\big]^{\mathrm{T}}, \tag{3}$$

where A is the following 3×3 matrix

$$A = \begin{bmatrix} 0 & \kappa(s) & 0 \\ -\kappa(s) & 0 & \tau(s) \\ 0 & -\tau(s) & 0 \end{bmatrix}. \tag{4}$$

The fundamental theorem of curves states as follows [10, 21]: Any regular curve of C^3 with $\kappa > 0$ is completely determined, up to position and orientation, by its curvature and torsion. In fact, given the total length of the curve segment, a starting point x_0 in $\mathbb{R}^3$, and an orthonormal basis which is the trihedron $[\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{B}]$ at x_0, the system of linear differential equations in (3) can be integrated and the space curve can be reconstructed. Thus, the pair $[\kappa(s), \tau(s)]$ can invariantly represent any space curve whose curvature is positive at every point on the curve. For planar curves, their torsion functions vanish. The curvature function of a planar curve is often calculated by the following formula

$$\kappa(s) = \left| \frac{\mathrm{d}\theta}{\mathrm{d}s} \right|,$$

where θ is the angle between the tangent vector at a point on the curve and the X coordinate axis.

In computer vision problems, the data of a space curve are generally available in the form of a list of 3-D points which are the sampled and digitized values of a piece-wisely smooth space curve function. Curve smoothing algorithms using regularization [28], cross validation [26], and smoothing kernels (Gaussian kernel [3] and B-spline kernel [11]) exist in the literature. It has been shown analytically that the variance of the positions of the breakpoints due to noisy data is inversely related to the support of the smoothing kernel [11]. In our experiments discussed later, we smooth the curves by B spline kernel, so that the smoothed versions of

the curves have sufficient differentiability. In the following, we shall present two representations for curves. One uses the differential geometric descriptors of their smoothed version. The other uses the invariant functions of the 3-D moments of the space curves.

Complex Waveform Representation

We propose an Euclidean invariant representation for general space curves. Our invariant representation is a complex waveform whose magnitude is the nonnegative curvature function and whose phase depends on its torsion function. The complex function can be defined at every point on the curve, even at those points whose curvatures vanish. The form of the phase function can be rather arbitrary. In this paper, we explicitly defined the complex function for a (smoothed) space curve as

$$I(s) = \kappa(s) \exp\left(j 2\tan^{-1} h(s)\right). \tag{5}$$

$I(s) = 0$ whenever $\kappa(s) = 0$; $h(s)$ in (5) is called helicity, which is defined at every point where $\kappa(s) > 0$, by

$$h(s) = \frac{\tau(s)}{\kappa(s)}. \tag{6}$$

The fundamental theorem of the space curves ensures that every space curve is uniquely represented by such a complex waveform. The waveform $I(s)$ is invariant under rotation and translation of coordinate system. Under space inversion, i.e., $\boldsymbol{x} \rightarrow -\boldsymbol{x}$, $\kappa(s)$ remains unchanged but $\tau(s)$ and $h(s)$ change their signs. Thus the torsions and helicities are pseudoscalar functions. $I(s) = 0$ for a straight line. When the curve is planar, the invariant waveform is simply the real nonnegative curvature function. The helices are nonplanar curves and have constant helicity function, so their waveforms have constant phases.

Many sophisticated signal processing algorithms require multiresolution symbolic descriptions of the signals. The scale maps or fingerprints are such descriptions [30]. We can construct the scale map of a space curve by first passing the complex waveform of the curve or the derivatives of the waveform, denoted by $J(s)$, through a Gaussian filter

$$E(s,\sigma) = G(s,\sigma) * J(s) = \left(\frac{1}{\sigma}\right) \int J(u) \exp\left(\frac{-(s-u)^2}{2\sigma^2}\right). \tag{7}$$

The scale map is the zero-crossing contour $s(\sigma)$ of the complex filtered signal $E(s,\sigma)$ on the s–σ plane. The filtered signal $E(s,\sigma)$ obeys the diffusion equation

$$\frac{\partial^2 E}{\partial s^2} = \frac{\partial E}{\partial t}, \quad t = \frac{\sigma^2}{2} \tag{8}$$

so that the t derivative can be expressed as the s derivative. The fingerprint theorem [31, 32] is crucially important in determining how much information of a signal

is stored in the scale map. The theorem ensures that the derivatives of the zero-crossing contours at two distinct points at the same scale, uniquely determine a signal up to a constant scale [31]. The proof of fingerprint theorem for a 1-D real signal consists of three steps [32]. We present here a brief proof of the fingerprint theorem for the complex $E(s,\sigma)$, following the same three steps. The first step is to show that the derivative at a point on a zero-crossing contour, i.e., $\partial^n E/\partial\xi^n = 0$ where ξ parametrizes the zero-crossing contour, constraints on the "complex moments" M_n of the Fourier transform of $E(s,\sigma)$

$$M_n\left(t=\sigma^2/2\right)=\int_{-\infty}^{\infty}(j\omega)^n \mathrm{e}^{-\omega^2 t}\mathrm{e}^{j\omega s}\hat{J}(\omega)\,\mathrm{d}\omega=\frac{\partial^n E}{\partial s^n}, \tag{9}$$

where $\hat{J}(\omega)$ is the Fourier transform of $J(s)$. In fact, the n equations $\partial^n E/\partial\xi^n = 0$ are a set of homogeneous linear equations of the first $2n$ moments. The second step relates the "complex moments" to the coefficients in the expression of $E(s,\sigma)$ in Hermite polynomial. Explicitly, $J(s)$ can be expanded as

$$J(s)=\sum_{n=0}^{\infty} a_n(\sigma)\phi_n(s,\sigma), \tag{10}$$

where $\phi_n(s,\sigma)$ is the following Hermite function

$$\phi_n(s,\sigma)=\frac{\sigma^{2n-1}}{n!\sqrt{2\pi}}\exp\left(\frac{s^2}{2\sigma^2}\right)\frac{\mathrm{d}^n}{\mathrm{d}s^n}\exp\left(\frac{-s^2}{2\sigma^2}\right).$$

The complex coefficients $a_n(\sigma)$ are related to the "complex moments" by

$$a_n(\sigma)=(-1)^n M_n(\sigma).$$

The third step is to show that it is generally possible to use the derivatives of another point on the zero-crossing contour to obtain n additional and independent equations for the "complex moments" such that $M_n(\sigma)$ and $a_n(\sigma)$ can be determined. The original signal can be reconstructed from (10). Therefore, we conclude that the scale-space image for 3-D curves can be derived and the fingerprint theorem holds true for the complex waveforms.

Moment Invariant Representation

3-D geometric moments of a space curve are defined by the following integral

$$M_{lmn}=\frac{1}{L}\int_0^L x^l(s)y^m(s)z^n(s)\,\mathrm{d}s, \tag{11}$$

where L is the total length of the curve. There are totally $(p+1)(p+2)/2$ moments of order p $(p = l+m+n)$. Since the central moments are invariant under translation, we shall use features which are rotation invariant functions of them. We denote the central moments by $\overline{M}_{lmn}$. These scalar moment functions can be

used as features for curve recognition. Three second order 3-D moment invariants are the eigenvalues of the following moment matrix of the curve [24],

$$\begin{bmatrix} \overline{M}_{200} & \overline{M}_{110} & \overline{M}_{101} \\ \overline{M}_{110} & \overline{M}_{020} & \overline{M}_{011} \\ \overline{M}_{101} & \overline{M}_{011} & \overline{M}_{002} \end{bmatrix}.$$

Higher order moments represent the fine spatial details on the curve and should be included in the feature vector for complete description. We have proposed a group-theoretic technique to derive scalar moment functions from the compounds of 3-D geometric moments via Clebsch–Gordon expansion [16]. This method is systematic and general. Explicit expressions of some invariant functions containing the second-order and the higher order moments are given in the Appendix. Moment calculation does not require estimation of derivatives from the noisy digitized data. Therefore, moment features are more robust for pattern classification. Experimental study has shown that the moment invariant features have very high discriminative power for noisy data.

Important feature points of a curve can be extracted when the curve is segmented into smooth segments. Kehtarnavaz et al. [11] used the magnitude of the Darboux vector, $|\Omega(s)| = (\kappa^2(s) + \tau^2(s))^{1/2} = \kappa(s)\sqrt{1 + h^2(s)}$, as the criterion function for 3-D curve segmentation. The breakpoints determined by their criterion function may correspond to the first- or the second-order discontinuities of the space curve. This criterion function for segmentation can be elegantly expressed as $I'(s) = 0$, if we define the following complex waveform for the space curve

$$I(s) = \kappa(s) \exp\left(j 2\tan^{-1}\sqrt{1 + h^2(s)}\right). \tag{12}$$

The waveform defined in (12) works just as well in curve matching as that defined in (5), even though the waveform is imaginary for planar curves. Nevertheless, the first order discontinuities of the curve can be more reliably detected and are more significant, because they are the control points for spline reconstruction of the curve. In our experiments, we simply define breakpoints to be the curvature maxima. In our breakpoint extraction, $\kappa'(s)$ of the smoothed curves are analyzed over a range of spatial scales. Breakpoints found at multiple scales are located at the finest scale. Geometric 3-D moments of the list of breakpoints $\{(x_i, y_i, z_i),\ i = 1, \ldots, N\}$ are defined as

$$M_{lmn} = \sum_{i=1}^{N} x_i^l y_i^m z_i^n. \tag{13}$$

A curve can be represented by the moment invariants of its breakpoints.

3. Matching and Recognition of Space Curves

In this section, we shall present algorithms for space curve matching using invariant global features which are either extracted from the complex waveforms or derived

from 3-D moments. The curves can be reliably identified by their global features when occlusion does not occur. The arc lengths of the curves depend, to some extent, on the orientations of the curves, because different amounts of digitizing noise are introduced into the curves at different orientations. The total length of the curve is thus a random variable and will be considered as one of the components in the global feature vector. The feature space can be partitioned into regions by the discriminant functions [9]. The data curve will be identified with a prototype, if its feature vector falls in the region of that prototype. We found in the experiments that the minimum distance classifier is suitable for space curve classification. Once a data curve has been identified with a model curve, the relative alignment and Euclidean transformation can then be determined by procedures described later in this section. The Euclidean transformation is not only useful in object positioning, but also useful as the rigidity constraint for integrating the individual curve matchings into a globally consistent matching, when the object is represented by a set of 3-D curves. In the following, we shall discuss the global feature extraction from the waveforms of the closed contours and the open 3-D curves.

3.1. GLOBAL FEATURES FROM THE INVARIANT WAVEFORM

3.1.1. *Fourier Descriptor*

The boundary contours of objects or smooth regions in the range data are closed curves. The invariant waveforms of closed contours are periodic functions of the arc length parameter. The period of each complex waveform is the total length of the corresponding contour. These periodic invariant functions can be expanded into Fourier series. The Fourier coefficients are defined as

$$c_n = \frac{1}{L}\int_0^L I(s)\exp\left(\frac{-j2\pi ns}{L}\right)\mathrm{d}s. \tag{14}$$

The data points on the boundary contours are usually extracted by contour following algorithms. Different starting points for contour tracing may be selected, when the boundary curve is extracted with the object in different orientations in the scene. The misalignment between the waveform of the curve in the scene, $I(s)$, and that of the corresponding curve on the model object, $I'(s)$ can be expressed as

$$I(s) = I'(s + s_0),$$

where s_0 is the offset. Their Fourier coefficients are related by

$$c_n = c_n' \exp\left(\frac{-j2\pi ns_0}{L}\right).$$

The magnitude of the Fourier coefficients are not affected by the choice of the starting point. Therefore, they can be chosen as the global features. We have found in our experimental study that, using five to eight Fourier features, it is possible to reliably classify various contours in the range data.

If the curve is planar, the invariant waveform becomes the curvature function of the curve, a smoother periodic function $a(s)$ can be defined as

$$a(s) = \int_0^s \kappa(t)\,\mathrm{d}t + \frac{2\pi s}{L}. \tag{15}$$

The Fourier features of planar curves are usually extracted from this function [34].

3.1.2. *Legendre Function Expansion*

Open space curve segments often appear in computer vision problems. Edges in the range data may not form closed curves. Moreover, the spine axes of the generalized cylinder representation of 3-D data are generally open curves. Global shape features of the open curves can be extracted from the orthogonal function expansion. The invariant waveform of an open space curve can be expressed as the following series

$$I(s) = \sum_{l=0}^{\infty} k_l \sqrt{l + \frac{1}{2}}\, P_l\left(\frac{2s}{L} - 1\right), \tag{16}$$

where $P_l(x)$'s are the Legendre functions which are orthogonal over $[-1, 1]$. In (16), we have expressed the argument of the Legendre function in terms of the arc length, $x = (2s/L) - 1$, so that these functions are orthogonal over the interval $[0, L]$. The complex coefficients k_l are defined by the integrals

$$k_l = \frac{2}{L}\sqrt{l + \frac{1}{2}} \int_0^L I(s)\, P_l\left(\frac{2s}{L} - 1\right) \mathrm{d}s. \tag{17}$$

The feature vector for an open space curve can be constructed from the real and imaginary parts of these coefficients.

3.2. COORDINATE TRANSFORM ESTIMATION

When a model curve entirely appears in the scene, the coordinate transformation can be calculated from the principal axes and the centroids of the curve in two coordinate frames [15, 16]. There is no need to establish exact point correspondence between two frames. However, this simple method is sensitive to noise. The least squares estimation method presented in the following is more robust. It works even when the curve is only partially visible in the scene. In 3-D space, the input data curve can be related to a corresponding segment of the model curve by an Euclidean transformation which minimizes the integral

$$\Delta(R, \boldsymbol{a}, s_0) = \int_0^L \left|R\boldsymbol{x}(s) + \boldsymbol{a} - \boldsymbol{y}(s + s_0)\right|^2 \mathrm{d}s, \tag{18}$$

where $\boldsymbol{x}(s)$ and $\boldsymbol{y}(s+s_0)$ are the data and the corresponding model curve segments, respectively, R is the rotation matrix, $\boldsymbol{a}$ is the translation vector, and s_0 is the offset

parameter. The integral can be expanded as

$$\begin{aligned}\Delta(R,\boldsymbol{a},s_0) \;=\; & \int_0^L |\boldsymbol{x}(s)-\boldsymbol{x}_0|^2\,\mathrm{d}s + A^{\mathrm{T}}AL + \int_0^L |\boldsymbol{y}(s+s_0)|^2\,\mathrm{d}s \\ & -2\int_0^L \boldsymbol{y}^{\mathrm{T}}(s+s_0)R(\boldsymbol{x}(s)-\boldsymbol{x}_0)\,\mathrm{d}s \\ & -2A^{\mathrm{T}}\int_0^L \boldsymbol{y}(s+s_0)\,\mathrm{d}s, \end{aligned} \tag{19}$$

where

$$A=\boldsymbol{a}+R\boldsymbol{x}_0, \qquad \boldsymbol{x}_0=\frac{1}{L}\int_0^L \boldsymbol{x}(s)\,\mathrm{d}s. \tag{20}$$

To minimize (19) over A, we get

$$A=\frac{1}{L}\int_0^L \boldsymbol{y}(s+s_0)\,\mathrm{d}s. \tag{21}$$

Substitute (21) into (19), we obtain

$$\begin{aligned}\Delta(R,\boldsymbol{a},s_0) \;=\; & \int_0^L \big|\boldsymbol{x}(s)-\boldsymbol{x}_0\big|^2\,\mathrm{d}s + \int_0^L \big|\boldsymbol{y}(s+s_0)\big|^2\,\mathrm{d}s \\ & -\frac{1}{L}\left|\int_0^L \boldsymbol{y}(s+s_0)\,\mathrm{d}s\right|^2 - 2\mathrm{Tr}\big(R\Sigma_{xy}(s_0)\big), \end{aligned} \tag{22}$$

The cross-correlation matrix is defined to be

$$\Sigma_{xy}(s_0)=\int_0^L (\boldsymbol{x}(s)-\boldsymbol{x}_0)\boldsymbol{y}^{\mathrm{T}}(s+s_0)\,\mathrm{d}s. \tag{23}$$

The rotation matrix that maximizes the trace term in (22), will minimize the integral Δ. To find such rotation matrix, we apply the singular value decomposition to $\Sigma_{xy}(s_0)$

$$\Sigma_{xy}(s_0)=U(s_0)\Lambda(s_0)V^{\mathrm{T}}(s_0), \tag{24}$$

where $\Lambda(s_0)$ is the diagonal singular value matrix, $V(s_0)$ and $U(s_0)$ are the orthogonal matrices. The rotation matrix

$$R(s_0)=V(s_0)U^{\mathrm{T}}(s_0) \tag{25}$$

will maximize the trace term. The maximum value of $\mathrm{Tr}(R\Sigma_{xy}(s_0))$, is $\mathrm{Tr}(\Lambda(s_0))$. Once $R(s_0)$ is determined, the translation vector $\boldsymbol{a}(s_0)$ can be calculated as

$$\boldsymbol{a}(s_0)=R\frac{1}{L}\int_0^L \boldsymbol{x}(s)\,\mathrm{d}s-\frac{1}{L}\int_0^L \boldsymbol{y}(s+s_0)\,\mathrm{d}s. \tag{26}$$

Both the rotation matrix and the translation vector are functions of the offset parameter s_0. Moreover, the point correspondence between $\boldsymbol{x}(s)$ and $\boldsymbol{y}(s)$ is also determined by the value of s_0. The offset parameter s_0 is the value which minimizes the following quantity

$$\Delta_{\min}(s_0) = \int_0^L \left|\boldsymbol{x}(s) - \boldsymbol{x}_0\right|^2 \mathrm{d}s + \int_0^L \left|\boldsymbol{y}(s+s_0)\right|^2 \mathrm{d}s - \frac{1}{L}\left|\int_0^L \boldsymbol{y}(s+s_0)\,\mathrm{d}s\right|^2 - 2\mathrm{Tr}\big(\Lambda(s_0)\big). \tag{27}$$

A similar approach of parameter estimation can be found in [5, 25], where the authors used polar decomposition to factor the Σ_{xy} into a product of orthogonal matrix and a nonnegative definite symmetric matrix [25]. The rotation matrix that maximizes $\mathrm{Tr}(R\Sigma_{xy}(s_0))$ is the transpose of the orthogonal matrix in the polar decomposition.

The rotation matrix can not be unambiguously determined when the cross correlation matrix is degenerate. The cross-correlation matrix of two straight lines is degenerate because it has two zero singular values. Straight lines have vanishing waveform, so they can be easily detected before using the least squares estimation. The relative orientation of two straight lines can be determined from their tangent vectors up to a sign ambiguity. The algorithm of matching 3-D line segments and its application to multiple-object recognition and motion estimation are discussed in [8].

When we implement our algorithm on a digital computer, the integrals in (27) will be approximated by discrete sums and evaluated at every sampled value of s_0. Each element in the cross-correlation matrix can be calculated by two FFT and one inverse FFT. Each FFT takes $\mathrm{O}(M \log_2 M)$ operations. Singular value decomposition of a $n \times n$ matrix takes $\mathrm{O}(n^3)$ operations. Decomposition of $\Sigma_{xy}(s_0)$ takes $12 \times 3^3 = 324$ flops. For $M = 256, 512$, the total matrix operations for the whole sequence $\Delta(s_0), s_0 = 0, 1, 2, \ldots, M-1$ have more or less the same number of computations as that for the evaluation of the cross-correlation matrix. By introducing the complex waveform matching, the computation complexity of the searching for s_0 can be reduced. If the model curve segment which starts at $\boldsymbol{y}(s_0)$ and ends at $\boldsymbol{y}(L+s_0)$ is the Euclidean transformed image of the data curve $\boldsymbol{x}(s)$, then the invariant shape waveforms of the two curves will be exactly the same. The offset parameter s_0 which minimizes (18) is one of the zeroes of the functions $D1(s_0)$ or $D2(s_0)$ defined by

$$D1(s_0) = \int_0^L \left|I_1(s) - I_2(s+s_0)\right|^2 \mathrm{d}s, \tag{28}$$

$$D2(s_0) = \int_0^L \left|I_1(L-s) - I_2(s+s_0)\right|^2 \mathrm{d}s, \tag{29}$$

where $I_1(s)$ and $I_2(s)$ are invariant shape waveforms of $\boldsymbol{x}(s)$ and $\boldsymbol{y}(s)$, respectively. $\Delta(s_0)$ is evaluated at each zero of $D1(s_0)$ or $D2(s_0)$ and the offset parameter s_0

is the one that predicts the smallest value of $\Delta(s_0)$. In the noisy curve data, we search for s_0 in the neighborhood of every pronounced local minima of $D1(s_0)$ or $D2(s_0)$. We then compute $\Sigma_{xy}(s_0)$, $\Delta(s_0)$ and $R(s_0)$ with $O(M)$ operations. However, the rotation matrices calculated from the cross-correlation matrix of mismatched curves are generally improper orthogonal matrices [2]. By checking the determinants of rotation matrices at various local minima, we can quickly find the offset parameter. $D1(s_0)$ and $D2(s_0)$ are dominated by the correlation terms

$$\text{Re} \int_0^L I_1(s) I_2^*(s + s_0)\, \mathrm{d}s \quad \text{and} \quad \text{Re} \int_0^L I_1(L - s) I_2^*(s + s_0)\, \mathrm{d}s,$$

respectively. They can be calculated with complexity of the order $O(M \log_2 M)$.

4. Matching Partially Occluded Curves

When one object is occluded by another object, the extracted boundary curves will contain boundaries from both objects. The global features of the boundary curves can not be used for classification. We may have to divide the boundary curve into segments, and match each segment to the curves of the model objects. In general, the scene may contain many discontinuous pieces of curves from several model objects. We shall determine the largest portion of each model curve existing in the scene and its position and orientation relative to its standard position. Our approach is to decompose a given data curve into a set of curve segments $\{x_i(s)$'s$\}$. Each $x_i(s)$ is used as a template and matched to various portions of each model curve $y_j(s)$ whose length is not shorter than that of $x_i(s)$. The list of breakpoints will guide the decomposition of data curves into templates. The intersection of boundaries of two objects will generally be a breakpoint. Therefore, templates can be the curve segments between breakpoints. The measure of the mismatch is the expression given in (18), or its equivalent in (27). Appropriate subscripts must be introduced to those equations, because each piece $x_i(s)$ on the data curve will match to every model curve $y_j(s)$ in the library. Ambiguities often occur when a template can locally match to several model curves, or to various portions of a model curve. These ambiguities can be resolved when we apply the rigidity constraint on pairwise curve segments. Moreover, the longest segment of a model curve in the scene can be determined, when the maximal set of mutually compatible matches to that model curve is found. The Euclidean transformations of these matches calculated from (25) and (26) will determine their structural compatibilities. To find the maximal set of mutually compatible matches, an association graph is constructed as follows. The node of the association graph is the pair of the matched curve segments $(x_i(s), y_j(s_0, s_0 + L_i))$, where $x_i(s)$ is a template and $y_j(s_0, s_0 + L_i)$ is a segment of the jth model curve with end points at $y_j(s_0)$ and $y_j(s_0 + L_i)$. Two nodes are connected by an edge, if the two templates in them match to the same model curve, and the relative Euclidean transformation parameters of these two matches are identical. The maximal cliques in the association graph are then

determined. Each clique is associated with only one model curve. The part of the model curve which appears in the scene is the union of all the templates in its clique. Since the computation of the translation vectors depends on the estimated rotation matrices, the estimation error of the rotation matrix will propagate to the estimated translation vector. Therefore, we usually give a relatively larger tolerance to the difference between translation vectors. We define the "distances" of rotation matrices and translation vectors by the following formulae

$$d(R_1, R_2) = \mathrm{Tr}\big((R_1 - R_2)^{\mathrm{T}}(R_1 - R_2)\big) < \varepsilon_R, \tag{30}$$

$$\big|\boldsymbol{a}_1 - \boldsymbol{a}_2\big| < \varepsilon_T. \tag{31}$$

The above template/structural algorithm can be used to recognize objects and interpret scenes which are represented by a set of 3-D curves. In these cases, the association graph generally contains a large number of nodes. The maximal cliques can be found by recursive procedures [4, 7]. They have the worst case complexity of exponential time. The template/structural curve matching algorithm may require a large amount of computation. The cluster growing approach in [12] finds a set of consistent matches with $O(n^2)$ time complexity, where n is the number of nodes in the graph. However, this approach can not guarantee that the resulting set is a maximal clique. The computation of curve matching can be reduced if we divide each data curve into long curve segments, and the attempted match is performed by matching invariant shape waveforms of the corresponding curve segments. We use long curves in waveform matching, because they can have significant shape characteristics. The waveform matching will be able to more efficiently find the corresponding curve segment on the model curve, and their relative offset parameter. In this case, the association graphs will contain relatively smaller numbers of nodes, and the cliques can be more quickly found.

Turney et al. proposed a template matching algorithm of 2-D curves using θ–s waveforms [29]. Our algorithm can be considered as a 3-D extension of their approach. Turney defined the saliency of each segment on the template, although the structural compatibility of matches was not emphasized in their paper. Based on our proposed complex waveform representation of space curves, other 2-D curve matching algorithms also have their 3-D extensions. The authors in [19] proposed a scale-space description of planar curves. The scale space image is a hierarchical representation and the zero-crossing contours can be structured into a tree. The matching of curves is to match their scale space images. This algorithm can be used to recognize partially visible curves. Since we have shown that the scaling theorem [33] and the fingerprint theorem are true for the scale maps of 3-D curves, the algorithm in [19] can be straightforwardly extended to 3-D curve matchnig.

5. Experimental Results

In the following, we present the computer simulation results and experimental study of our curve matching algorithm. We first tested our algorithm using curves

Table I. Means and standard deviations of Fourier features extracted from the invariant waveforms of curves A and B, respectively

Fourier features	Mean (A)	Stand. dev. (A)	Mean (B)	Stand. dev. (B)
c_0	0.153	0.0	0.177	0.0
c_1	0.151	2.0×10^{-5}	0.175	1.2×10^{-5}
c_2	0.137	1.5×10^{-4}	0.164	8.0×10^{-5}
c_3	0.116	2.0×10^{-5}	0.150	0.1×10^{-5}
c_4	0.107	1.7×10^{-5}	0.146	0.8×10^{-5}
c_5	0.098	1.4×10^{-5}	0.134	1.5×10^{-5}
c_6	0.092	0.2×10^{-5}	0.126	0.8×10^{-5}

which can be described by parametric equations. In the first experiment, the selected closed curves can be described by

$$x(t) = r_x \sin\theta(t) \cos\phi(t),$$
$$y(t) = r_y \sin\theta(t) \sin\phi(t),$$
$$z(t) = r_z \cos\theta(t),$$

where

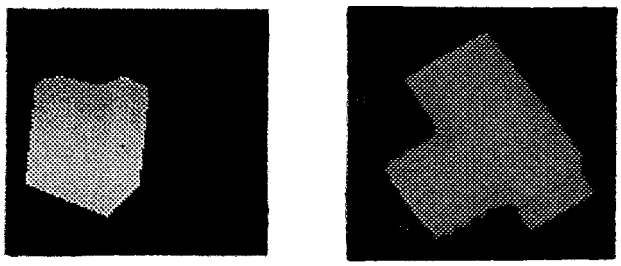
$$\theta(t) = 2\pi t^2, \qquad \phi(t) = 2\pi t, \quad 0 \leqslant t \leqslant 1.$$

Curves A and B were generated by the equations using parameters $r_x^A = 30$, $r_y^A = 30$, $r_z^A = 50$ and $r_x^B = 30$, $r_y^B = 40$, $r_z^B = 50$, respectively. The curves were rotated to 216 different orientations. At each orientation, they were uniformly sampled, and the Fourier features of the invariant waveforms of these curves were computed. The statistical parameters of these Fourier features are shown in Table I. As can be seen, these features are essentially invariant under rotation and translation. We have also tested our algorithms using the synthetic curves generated by Hermite splines, which are open curves. The tangents of the curves were made discontinuous at certain points. The locations of these breakpoints were determined by tracking the curvature maxima from the coarse to fine scales. The Legendre features k_n, as defined in (17), were extracted from these open space curves. The invariance of these features have been justified in the experiments. In the following experiment, the spiral curves were used, which can be described by the following equations

$$x(t) = re^{bt} \cos(ft),$$
$$y(t) = re^{bt} \sin(ft),$$
$$z(t) = re^{bt},$$

Table II. Means and standard deviations of the real and the imaginary parts of the Legendre features, extracted from the open space curve $x(t) = re^{bt}\cos(ft)$, $y(t) = re^{bt}\sin(ft)$, $z(t) = re^{bt}$

Legendre features	Mean (real)	Stand. dev. (real)	Mean (imag.)	Stand. dev. (imag.)
k_0	1.21×10^{-2}	3.0×10^{-6}	1.34×10^{-2}	2.27×10^{-6}
k_1	-1.51×10^{-2}	3.35×10^{-6}	-1.69×10^{-2}	3.10×10^{-6}
k_2	9.544×10^{-3}	2.647×10^{-6}	1.08×10^{-2}	2.27×10^{-6}
k_3	-2.86×10^{-3}	1.90×10^{-6}	-3.26×10^{-3}	1.74×10^{-6}
k_4	-7.41×10^{-5}	2.21×10^{-6}	-1.40×10^{-4}	2.00×10^{-6}
k_5	-2.68×10^{-4}	3.00×10^{-6}	-2.12×10^{-4}	2.72×10^{-6}



(a) (b)

Figure 1. Two range images.

where $0 \leqslant t \leqslant 2$, and $r = 10, b = 0.5, f = 5$. The statistical variances of the computed Legendre features due to digitizing errors are shown in Table II. As can be seen, they are also invariant under rotation and translation.

We then tested our algorithm using space curves determined by the depth discontinuity in the range images. The synthetic range image in Figure 1(a) was generated by a computer graphics technique. The range image in Figure 1(b) was obtained by a laser range scanner. The objects in these images were first segmented from the background. Their external boundaries were then extracted by a contour following procedure and were stored in one-dimensional arrays. The objects were rotated and translated to various positions in a way such that the same views were preserved, i.e., the visible portion of the objects were the same except that the viewing angles were different. Because of aliasing, the boundary contours of the objects look jagged. Also, the boundary contours, of each object in different orientations contain different numbers of pixels. The contours were smoothed by spline fitting. The Fourier features were then extracted from the smoothed boundaries of

Table III. Means and standard deviations of Fourier features extracted from the invariant waveforms of the external boundaries of the objects in Figures 1(a) and (b)

Fourier features	Mean (a)	Stand. dev. (a)	Mean (b)	Stand. dev. (b)
c_0	0.48	0.23	2.60	0.22
c_1	0.09	0.07	0.32	0.18
c_2	0.12	0.10	0.32	0.10
c_3	0.11	0.08	0.39	0.13
c_4	0.09	0.08	0.30	0.09
c_5	0.09	0.06	0.13	0.07
c_6	0.05	0.02	0.20	0.10

Table IV. Moment invariants evaluated from the boundaries in Figures 1(b) and 2

Moment invariants	Figure 2(a)	Figure 2(b)	Figure 2(c)	Figure 1(b)
I^2_{22}	0.159	0.159	0.159	0.150
I^2_{222}	–0.051	–0.501	–0.501	–0.046
I^3_{33}	–6.833	–7.760	–6.946	–1.036
I^3_{11}	–44.904	–46.558	–51.731	–8.937
$I^{2,3}_{233}$	2.074	2.271	2.129	0.304
$I^{2,3}_{123}$	–1.293	–1.347	–1.411	–0.218
$I^{2,3}_{112}$	3.381	3.486	3.898	0.652
I^3_{3333}	27.992	35.527	28.742	0.627
I^3_{1333}	–16.880	–19.722	–18.936	–0.442
I^3_{1133}	10.693	12.292	12.596	0.321
I^3_{1113}	–27.408	–28.800	–34.597	–0.941

the objects in various positions. The means and variances of these features were estimated and are shown in Table III. We constructed the maximum likelihood classifiers with Gaussian distribution using these statistical parameters and found that the classifiers could always unambiguously distinguish between these two boundary curves. To show the performance of the 3-D moment invariants for space curves as listed in Appendix, we computed them for the boundary contours of the synthetic object in various positions shown in Figure 2(a)–(c), and Figure 1(b). The results are shown in Table IV. Among the eleven 3-D invariant moment functions in Table IV, I^2_{22}, I^2_{222} contain the second-order moments; I^3_{11}, I^3_{33}, I^3_{3333}, I^3_{1333}, I^3_{1133},

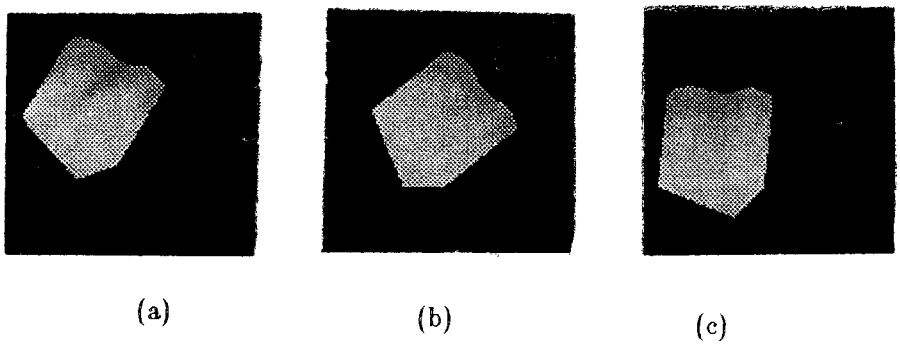


Figure 2. Range images of an object in various orientations.

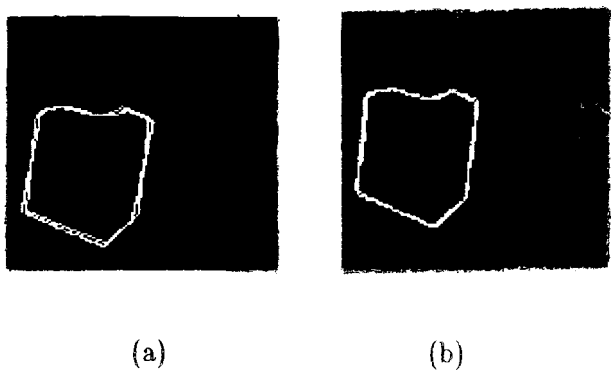
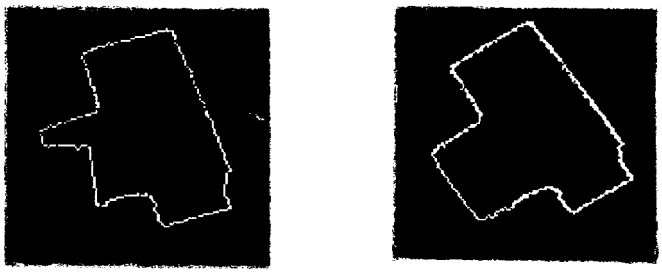


Figure 3. Overlapping of the transformed boundaries in Figures 2(a) and 2(b) on the boundary in Figure 2(c), respectively.

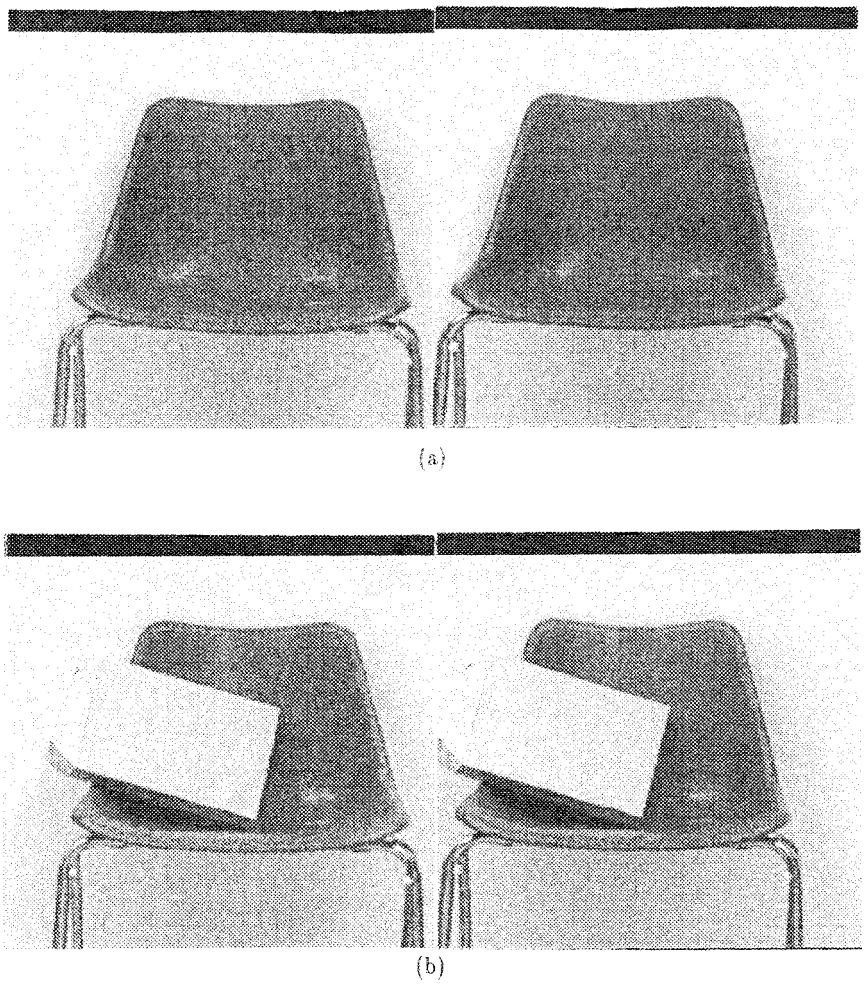
I^3_{1113} contain the third-order moments; and $I^{2,3}_{233}$, $I^{2,3}_{123}$, $I^{2,3}_{112}$, contain both the second- and the third-order moments. It is evident that these moment features are invariant to the 3-D rotations and translations. The point correspondence and motion parameters between these curves in Figure 2 were also calculated using the algorithm given in Section 3. The boundary contours of the images in Figures 2(a) and (b) were transformed by the estimated motion parameters and were overlapped on the boundary of the image in Figure 2(c). The results are shown in Figures 3(a) and (b), respectively. The range data in Figure 1(b) were also rotated to a different orientation, and 15% edge points were deleted from its boundary by occlusion, as shown in Figure 4(a). This partially obscured contour was matched to the boundary of the image in Figure 1(b) following the procedure given in Section 4. The result is shown in Figure 4(b).

3-D data of curves can also be obtained by stereopsis. We present in the following the experimental results of curve matching using 3-D data extracted from the stereo images. The first scene contains a single chair whose stereo images are shown in Figure 5(a). The second scene contains two chairs whose stereo images



(a) (b)

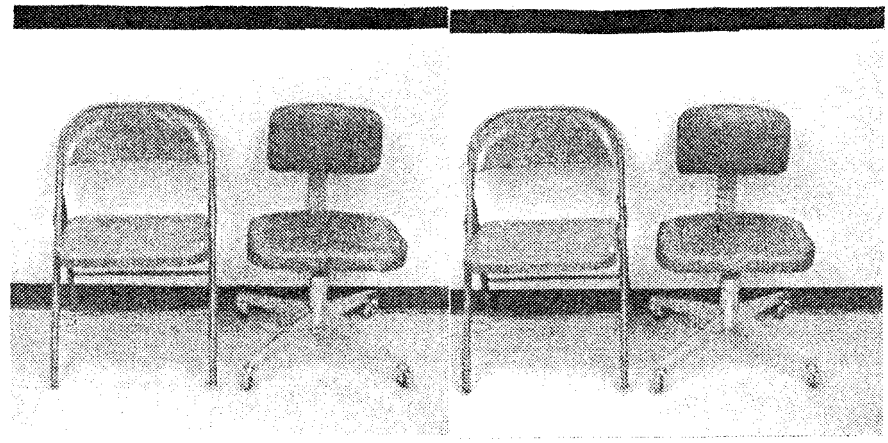
Figure 4. (a) Boundary of a partially occluded object in Figure 1(b). (b) Overlapping of the matched boundaries in Figures 4(a) and 1(b).



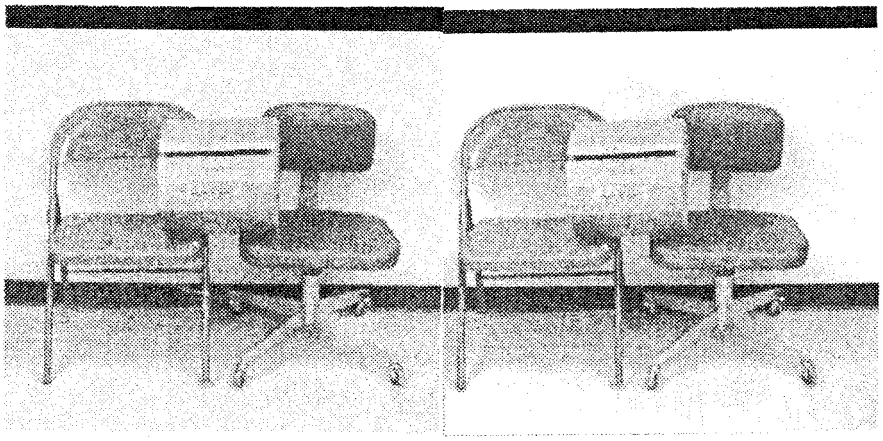
(a)

(b)

Figure 5. (a) Stereo images of a chair, and (b) stereo images of the occluded chairs.



(a)



(b)

Figure 6. (a) Stereo images of two chairs, and (b) stereo images of the occluded chairs.

are shown in Figure 6(a). The chairs were then occluded by paper boxes. The stereo images of the occluded scenes are shown in Figures 5(b) and 6(b), respectively. The images were first adaptively thresholded into binary images. The large regions in the thresholded images were then labeled. The boundaries of the large regions were traced, smoothed and stored, which formed the characteristic curves of the scene. The curves in the occluded scenes were used as data curves, and the curves in the non-occluded scenes were used as model curves. There were one model curve and five templates in the first scene, three model curves and ten templates in the second scene, which are shown in Figures 7(a) and 7(b), respectively. The epipolar line is along the X-axis. We computed the disparity map of the chairs by a correlation-based stereo matching algorithm. The disparity map contains the information of

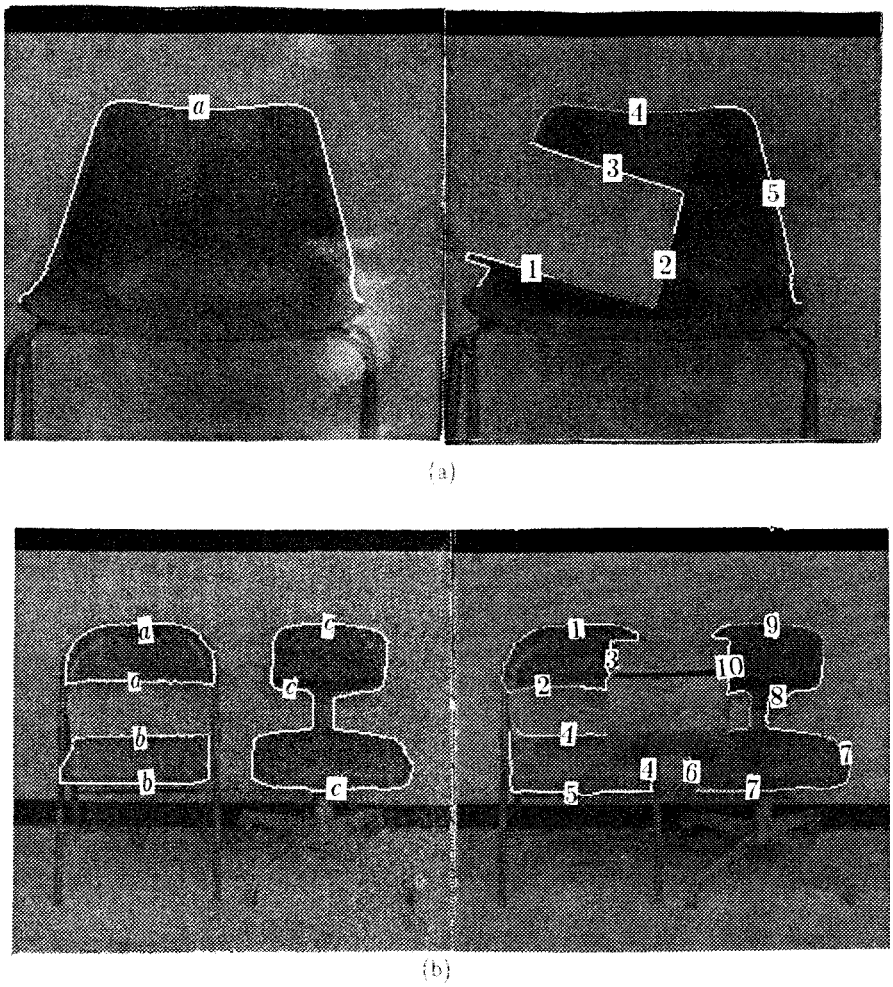


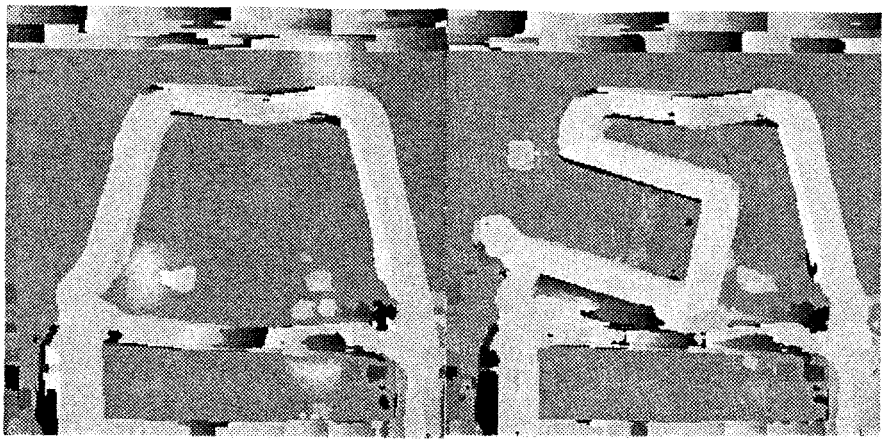
Figure 7. Labeled model curves (left) and the templates of unknown objects (right) of the first scene and the second scene, respectively.

point correspondence between the left and right images. The disparity map is in one-to-one correspondence to the right image. The grey value at each pixel is a linear function of the disparity value at that pixel. The 3-D coordinates of the boundary curves were computed from the formula

$$x = b\frac{(\text{disparity} + 2x_r)/2}{\text{disparity}},$$

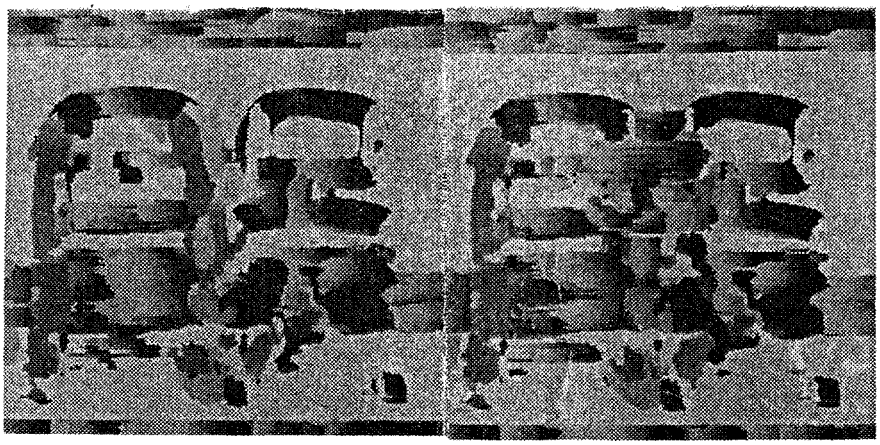
$$y = b\frac{(y_l + y_r)/2}{\text{disparity}},$$

$$z = \frac{bf}{\text{disparity}},$$



(a) (b)

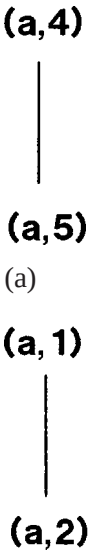
Figure 8. Disparity maps computed from Figures 5(a) and (b), respectively.



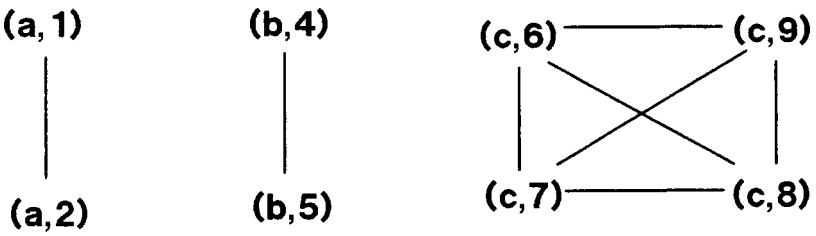
(a) (b)

Figure 9. Disparity maps computed from Figures 6(a) and (b), respectively.

where the focal length f and base line b for the images in Figure 5 are $f = 35$ mm and $b = 70$ mm. For the images in Figure 6, they are $f = 17.5$ mm and $b = 35$ mm. The generated disparity maps are shown in Figures 8 and 9, respectively. As can be seen, they are very noisy. Disparity might be undetermined at some points along the boundary curves. A particular grey value, 132, was given to that point of the curve. The assumption of slowly varying disparity was used for interpolation. The disparity map in Figure 9(b) has large regions of undetermined disparities. The 3-D coordinates of the curve segments in those regions could not be calculated and the segments were discarded. Some templates, such as those in

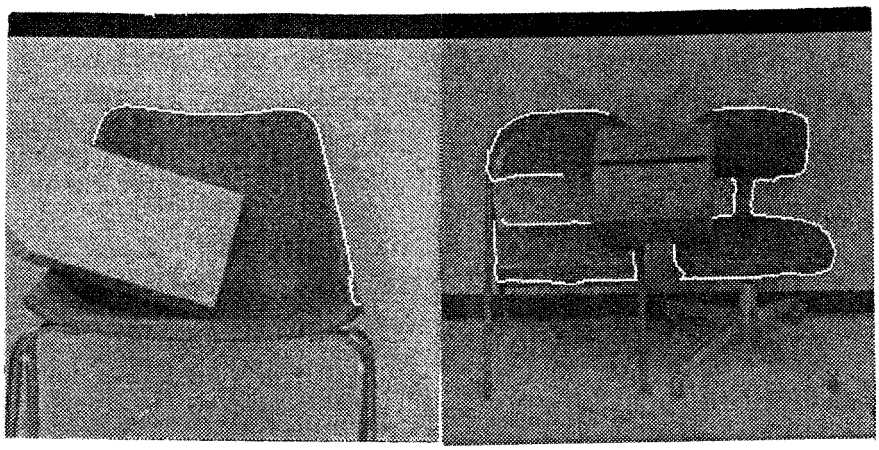


(a)



(b)

Figure 10. The consistent matches between the templates of unknown objects and the model curves for the first scene and the second scene, respectively.



(a) (b)

Figure 11. The visible part of the model curves predicted by the curve matching algorithm for the first scene and the second scene, respectively.

Figure 7(b), are discontinuous. The templates were matched to the models using the algorithm described in Section 4. The offset parameter s_0 and coordinate transformations of the matches were determined. There are two consistent matches in the first scene, eight consistent matches represented by three cliques in the association graph in the second scene. These graphs are shown in Figure 10. The visible model curves predicted by our curve matching algorithm are displayed in Figure 11. We can see that our algorithm of curve matching works satisfactorily.

6. Conclusions

In this paper, we have proposed an invariant waveform representation for space curves. We have also discussed the problem of space curve matching. The proposed invariant representation contains the complete 3-D shape information of a space curve. This invariant representation allows us to manipulate the segmentation and recognition of 3-D curves in the same way as those of 1-D waveforms, which have been very well studied in the literature. Moment invariants of 3-D curves have also been discussed, which are very discriminative features for pattern classification. An algorithm using the invariant global features for matching the nonoccluded 3-D curves has been discussed. This approach is fast and robust. A template/structural algorithm has been proposed for the recognition of partially occluded curves. The waveform matching can be used for efficient template matching and point correspondence problems. All of our results can be used to describe and match 2-D curves. In our current research, we are trying to improve the result of curve smoothing and to scale the performance of template waveform matching. We are also developing a robust object recognition algorithm using curve matching, where the saliency of the features will be taken into account.

Acknowledgement

This work was supported by the National Science Council, Taiwan, ROC, under Grant NSC 84-2213-E-005-022. We thank Prof. Wei-Chung Lin in the Department of EECS, Northwestern University, USA, for providing the range image in Figure 1(b). We also thank Dr. Bill Sakoda for providing the software for computing the disparity maps in our experiments.

Appendix

A brief summary of some of the 3-D moment invariants are listed below, which were used in the experiments given in this paper.

$$I_{22}^2 = \frac{\nu(2,2)_0^0}{(\nu_0^0)^2}, \qquad I_{222}^2 = \frac{\eta\nu}{(\nu_0^0)^3}, \qquad I_{33}^3 = \frac{\nu(3,3)_0^0}{(\nu_0^0)^{12/5}},$$

$$I_{11}^{3} = \frac{\nu(1,1)_0^0}{(\nu_0^0)^{12/5}}, \qquad I_{233}^{2,3} = \frac{\nu(3,3)_2\nu_2}{(\nu_0^0)^{17/5}}, \qquad I_{123}^{2,3} = \frac{\nu(3,1)_2\nu_2}{(\nu_0^0)^{17/5}},$$

$$I_{112}^{2,3} = \frac{\nu(1,1)_2\nu_2}{(\nu_0^0)^{17/5}}, \qquad I_{3333}^{3} = \frac{\nu^2(3,3)_2}{(\nu_0^0)^{24/5}}, \qquad I_{1333}^{3} = \frac{\nu(3,3)_2\nu(3,1)_2}{(\nu_0^0)^{24/5}},$$

$$I_{1133}^{3} = \frac{\nu^2(3,1)_2}{(\nu_0^0)^{24/5}}, \qquad I_{1113}^{3} = \frac{\nu(3,1)_2\nu(1,1)_2}{(\nu_0^0)^{24/5}}.$$

The detailed definitions and derivations of the quantities $\nu(i,j)_m^n$'s appeared in the above expressions can be found in [16].

References

1. Agin, G. J. and Binford, T. O.: Computer description of curved objects, in: *Proc. of the 3rd Internat. Joint Conf. on Artificial Intelligence*, Stanford, CA, August 20–23, 1973, pp. 629–640.
2. Arun, K. S., Huang, T. S., and Blostein, S. D.: Least square fitting of two 3-D point sets, *IEEE Trans. Pattern Anal. Mach. Intell.* **9** (1987), 698–700.
3. Asada, H. and Brady, M.: The curvature primal sketch, *IEEE Trans. Pattern Anal. Mach. Intell.* **8** (1986), 2–14.
4. Ballard, D. H. and Brown, C. M.: *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
5. Bastuscheck, C. M. et al.: Object recognition by three-dimensional curve matching, *Int. J. Intell. Systems* **1** (1986), 105–132.
6. Besl, P. J. and Jain, R. C.: Three-dimensional object recognition, *ACM Comput. Surveys* **17**(1) (1985), 75–145.
7. Bolles, R. C. and Cain, R. A.: Recognizing and locating partially visible objects: The focus feature method, *Internat. J. Robotics Res.* **1**(3) (1982), 57–81.
8. Chen, H. H. and Huang, T. S.: An algorithm for matching 3-D line segments with application to multiple-object motion estimation, in: *Proc. of Workshop on Computer Vision*, Miami, FL, November 30–December 2, 1987, pp. 151–156.
9. Duda, R. O. and Hart, P. E.: *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
10. Hsiung, C. C.: *A First Course in Differential Geometry*, Wiley/Interscience, New York, 1981.
11. Kehtarnavaz, N. and de Figueiredo, R. J. P.: A 3-D contour segmentation scheme based on curvature and torsion, *IEEE Trans. Pattern Anal. Mach. Intell.* **10** (1988), 707–713.
12. Koch, M. and Kashyap, R.: Using polygons to recognize and locate partially occluded objects, *IEEE Trans. Pattern Anal. Mach. Intel.* **9** (1987), 483–494.
13. Kuhl, F. P. and Giardina, C. R.: Elliptic Fourier features of a closed contour, *Computer Graphics Image Processing* **18** (1982) 236–258.
14. Lin, C. S. and Hwang, C. L.: New forms of shape invariants from elliptic Fourier descriptors, *Pattern Recognition* **20** (1987), 535–545.
15. Lin, Z., Lee, H., and Huang, T. S.: Finding 3-D point correspondences in motion estimation, in: *Proc. the Eighth Int. Conf. on Pattern Recognition*, 1986, pp. 303–305.
16. Lo, C. H. and Don, H. S.: 3-D moment forms: Their construction and application to object identification and positioning, *IEEE Trans. Pattern. Anal. Mach. Intell.* **11** (1989), 1053–1064.
17. Lo, C. H. and Don, H. S.: Representation and recognition of 3-D curves, in: *Proc. CVPR*, 1989, pp. 523–528.
18. Mokhtarian, F.: Multi-scale description of space curves and three-dimensional objects, in: *Proc. CVPR*, 1988, pp. 298–303.
19. Mokhtarian, F. and Mackworth, A.: Scale-based description and recognition of planar curves and two-dimensional shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* **8** (1984), 34–43.

20. Nevatia, R. and Binford, T. O.: Description and recognition of complex-curved objects, *Artificial Intell.* **8** (1977), 77–98.
21. O'Neill, B.: *Elementary Differential Geometry*, Academic Press, New York, 1966.
22. Pavlidis, T. and Ali, F.: A hierarchical syntactic shape analyzer, *IEEE Trans. Pattern Anal. Mach. Intell.* **1** (1979), 2–9.
23. Pavlidis, T. and Horowitz, S. L.: Segmentation of plane curves, *IEEE Trans. Comput.* **23** (1974), 860–870.
24. Sadjadi, F. A. and Hall, E. L.: Three-dimensional moment invariants, *IEEE Trans. Pattern Anal. Mach. Intell.* **2** (1980), 127–136.
25. Schwartz, J. T. and Sharir, M.: Identification of partially obscured objects in two and three dimensions by matching noisy characteristics curves, *Internat. J. Robotics Res.* **6** (1987), 29–44.
26. Shahraray, B. and Anderson, D.: Optimal estimation of contour properties by cross-validated regularization, *IEEE Trans. Pattern Anal. Mach. Intell.* **11** (1989), 600–610.
27. Shapiro, L. G. and Haralick, R. M.: Decomposition of two-dimensional shape by graph theoretic clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* **1** (1979), 10–20.
28. Terzopoulos, D.: Regularization of inverse visual problems involving discontinuities, *IEEE Trans. Pattern Anal. Mach. Intell.* **8** (1986), 413–424.
29. Turney, J., Mudge, T., and Volz, R.: Recognizing partially occluded parts, *IEEE Trans. Pattern Anal. Mach. Intell.* **7** (1985), 410–421.
30. Witkin, A.: Scale-space filtering, in: *Proc. Internat. Joint Conf. on Artificial Intelligence*, 1983, pp. 1019–1022.
31. Yuille, A. and Poggio, T.: Fingerprints theorems, in: *Proc. Internat. Joint Conf. on Artificial Intelligence*, 1984, pp. 362–365.
32. Yuille, A. and Poggio, T.: Fingerprints theorems, *J. Optic. Soc. Amer.* **2**(5) (1985), 683–692.
33. Yuille, A. and Poggio, T.: Scaling theorems for zero crossings, *IEEE Trans. Pattern Anal. Mach. Intell.* **8** (1986), 15–25.
34. Zahn, C. T. and Roskies, R. Z.: Fourier descriptors for plane closed curves, *IEEE Trans. Comput.* **21** (1972), 269–281.